

# METHOD, APPARATUS AND PROGRAM FOR IMAGE COMPRESSION, AND IMAGE READING APPARATUS

Patent number: JP2006279741 (A)

Publication date: 2006-10-12

Inventor(s): MOMOZONO YUKINOBU

Applicant(s): SEIKO EPSON CORP

Classification:

- international: H04N1/41; G06T1/00; G06T9/00; H04N1/46; H04N1/60; H04N1/41; G06T1/00; G06T9/00; H04N1/46; H04N1/60

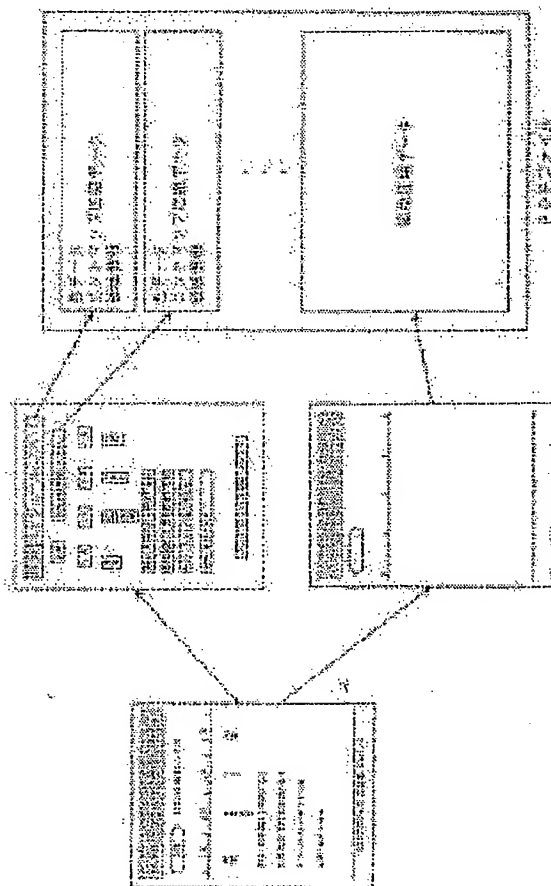
- european:

Application number: JP20050098337 20050330

Priority number(s): JP20050098337 20050330

## Abstract of JP 2006279741 (A)

**PROBLEM TO BE SOLVED:** To provide a method, apparatus and program for image compression and image reading apparatus with which a digital image is compressed with high compressibility so as not to damage readability of character.; **SOLUTION:** The image compression method includes: a setting stage of setting a processing target region on a digital image; a grouping stage of grouping pixels constituting the processing target region for each pixel with little color difference; a selecting stage of selecting a replacement target group based on feature amounts of the groups; a location information generating stage of indicating a location of each of the pixels belonging to the replacement target group; a replacing stage of replacing the color of each of pixels belonging to the replacement target group with the color of surrounding pixels; a digital image compressing stage of generating image compression data by compressing the digital image after the color of each of the pixels is replaced; and an output stage of outputting color data representing the representative color of the replacement target group, the location information and the image compression data in an associated manner.; **COPYRIGHT:** (C)2007,JPO&INPIT



(19) 日本国特許庁(JP)

## (12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-279741

(P2006-279741A)

(43) 公開日 平成18年10月12日(2006.10.12)

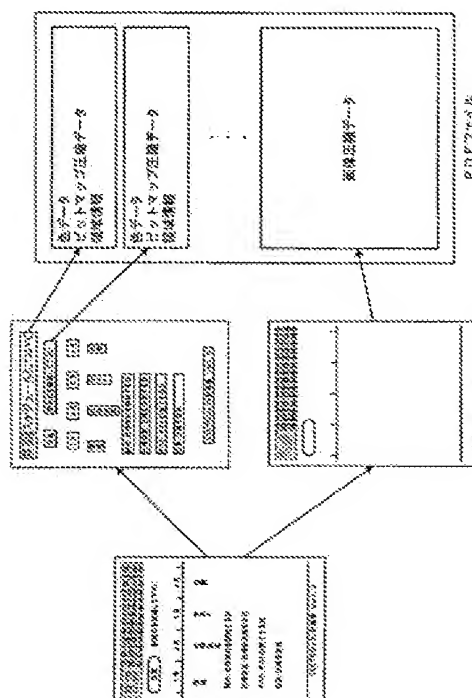
(51) Int. Cl.			F 1			テーマコード (参考)	
H04N	1/41	(2006.01)	H04N	1/41	C	SB057	
G06T	1/00	(2006.01)	G06T	1/00	S10	SC077	
G06T	9/00	(2006.01)	G06T	9/00		SC078	
H04N	1/60	(2006.01)	H04N	1/40	D	SC079	
H04N	1/46	(2006.01)	H04N	1/46	Z		

審査請求 未請求 請求項の数 12 O L (全 21 頁)

(31) 出願番号 特願2005-98337 (P2005-98337)  
 (22) 出願日 平成17年3月30日(2005.3.30)

(71) 出願人 000002369  
 セイコーエプソン株式会社  
 東京都新宿区西新宿2丁目4番1号  
 (74) 代理人 100093779  
 弁理士 服部 雅紀  
 (74) 代理人 100117396  
 弁理士 吉田 大  
 (72) 発明者 桃園 幸信  
 長野県松本市芳川村井町1059番地 株式会社エプソンソフト開発センター内  
 Fターム(参考) 5B057 CA01 CA08 CA12 CA16 CB18  
 CE10 CE17 CG05 DA12 DB02  
 DB06 DB09 DC25  
 5C077 LL19 MP01 MP04 PP23 PP36  
 PP37 PP68 PQ08 PQ12 RR21  
 最終頁に続く

(54) 【発明の名称】 画像圧縮方法、画像圧縮装置、画像圧縮プログラム及び画像読み取り装置



- <1> 本発明は、画像圧縮方法、画像圧縮装置、画像圧縮プログラム及び画像読み取り装置に関する。
- <2> 原稿を読み取ってデジタル画像を生成し、生成したデジタル画像を圧縮して着脱自在のメモリカードに保存する画像読み取り装置が知られている（例えば、特許文献１参照）。しかしながら、メモリカードはパーソナルコンピュータが備えるハードディスクなどに比べて一般に記憶容量が小さく、一方、デジタル画像は圧縮してもまだデータ量が大きいので、デジタル画像をメモリカードに保存するとメモリカードが直ぐに一杯になってしまう。
- <3> また、例えば文書を他人に配布する場合、紙の文書を画像読み取り装置で読み取ってデジタル画像を生成し、生成したデジタル画像を電子メールに添付して送付することがある。しかしながら前述したようにデジタル画像は圧縮してもまだデータ量が大きいので、電子メールに添付して送信すると送信に時間がかかったり送信エラーになって送信できなかったりする場合がある。
- <4> このようにデジタル画像のデータ量が大きいと様々な問題が生じるため、デジタル画像をより小さく圧縮できる方法が望まれている。ＪＰＥＧ圧縮は圧縮率が可変であるため、より高圧縮率で圧縮することでデータ量をより小さくすることができる。しかしながら、ＪＰＥＧ圧縮は不可逆圧縮であり、不可逆圧縮では圧縮率を高くするほど多くのデータが切り捨てられるので、圧縮率を高くするほど画質が低下する。例えば文書を読み取って生成したデジタル画像の場合は、圧縮率を高くすると再生の際に文字が滲んで読み難くなってしまう。このため文書を読み取って生成したデジタル画像の場合は安易に圧縮率を高くできないという問題がある。
- <6> 本発明は、上記問題に鑑みて創作されたものであって、デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮する画像圧縮方法、画像圧縮装置、画像圧縮プログラム及び画像読み取り装置を提供することを目的とする。
- <7> （１）上記目的を達成するため、本発明に係る画像圧縮方法は、デジタル画像上に処理対象領域を設定する設定段階と、前記処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化段階と、各前記グループの特徴量に基づいて置換対象グループを選択する選択段階と、前記置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成段階と、前記置換対象グループに属する各画素の色を前記置換対象グループに属さない周囲の画素の色で置換する置換段階と、前記置換段階で前記各画素の色が置換された後の前記デジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮段階と、前記置換対象グループの代表色を表す色データ、前記位置情報、及び前記画像圧縮データを関連付けて出力する出力段階と、を含む。
- 処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化した場合、文字を表す画素が属するグループはその特徴量が特定の傾向を持っていることが多い。従って、特徴量を用いれば文字を表す画素が属するグループを高い確率で特定することができる。文字を表す画素が属するグループを置換対象グループとし、デジタル画像において置換対象グループに属する画素の色を周囲の画素の色で置換すると、デジタル画像から文字を除去することができる。デジタル画像から文字を除去するとデジタル画像の圧縮率は文字の読み易さに影響を与えなくなるので、デジタル画像を高圧縮率で圧縮することができる。一方、文字を表す画素の位置を示す位置情報は各画素の濃度を持つ必要がないので小さなデータ量でよい。従ってデジタル画像全体のデータ量を小さくできる。再生の際にはデジタル画像において位置情報が示す画素の色を色データが表す代表色に置換することにより、文字は滲むことなく再生される。よって本発明によると、デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮できる。
- <8> （２）前記設定段階は、前記デジタル画像を複数のブロックに分割し、各前記ブロックを周波数成分に変換する変換段階と、各前記ブロックの高周波成分に基づいて前記処理対象領域を設定する領域設定段階と、を含んでもよい。
- 文字を含まない領域が処理対象領域に設定されると処理に無駄が生じる。文字を含む領域は周波数成分に変換すると高周波成分が大きな値となるので、各ブロックの高周波成分に基づいて処理対象領域を設定すると、文字を含まない領域が処理対象領域に設定されることによる無駄を低減できる。
- <9> （３）前記領域設定段階は、前記デジタル画像全体を前記処理対象領域として設定する段階と、前記処理対象領域について前記ブロックからなる行毎に高周波成分が所定値以上である前記ブロックの数を集計し、高周波成分が所定値以上である前記ブロックの数が第一の閾値未満の行で前記処理対象領域を分割する段階と、前記処理対象領域について前記ブロックからなる列毎に高周波成分が所定値以上である前記ブロックの数を集計し、高周波成分が所定値以上である前記ブロックの数が第二の閾値未満の列で前記処理対象領域を分割する段階とを含み、行で前記処理対象領域を分割する前記段階と列で前記処理対象領域を分割する前記段階とを繰り返すことにより前記ディ

デジタル画像上に複数の矩形の前記処理対象領域を設定してもよい。

複数の処理対象領域を設定すると、特徴量が特定の傾向をより顕著に持つようになるので、文字を表す画素が属するグループをより確実に選択することができる。

<10> (4) 前記グループ化段階において、前記処理対象領域を構成する画素を *k-means* 法によってグループ化してもよい。

*k-means* 法によってグループ化すると、画素をより適切にグループ化できる。

<11> (5) 前記特徴量は前記グループに属する画素の数であり、前記選択段階において、画素の数が二番目に多い前記グループを前記置換対象グループとして選択してもよい。

文書を読み取って生成したデジタル画像では、一般に背景を表す画素の数が最も多く、文字を表す画素の数は二番目に多い。従って画素の数が二番目に多いグループを選択することにより、文字を表す画素が属するグループを選択できる。

<12> (6) 他の前記グループに属する画素に周囲を囲まれて同一の前記グループに属する他の画素と隔離されている画素群を小グループというとき、前記特徴量は前記グループに属する前記小グループの数であり、前記選択段階において、前記小グループの数が最も多い前記グループを前記置換対象グループとして選択してもよい。

一般に文字はいくつかの互いに離間した直線や曲線で構成されている。すなわち文字はいくつかの小グループで構成されている。従って小グループの数が最も多いグループを選択することにより、文字を表す画素が属するグループを選択できる。

<13> (7) 前記選択段階において、複数の前記グループを前記置換対象グループとして選択してもよい。

文字が2色以上で表されている場合、複数のグループを選択することで、文字を表す画素が属するグループが置換対象グループとして選択されずにデジタル画像に文字が残ってしまうことを低減できる。

<14> (8) 前記位置情報は、前記処理対象領域を構成する画素を前記置換対象グループに属する画素と前記置換対象グループに属さない画素とに二値化した二値画像、及び前記置換対象グループが属する前記処理対象領域の位置を特定するための領域情報であってもよい。

<15> (9) 前記出力段階は、前記色データ、前記位置情報、及び前記画像圧縮データを一つのPDFファイルに格納して出力してもよい。

PDFファイルを再生するプログラムは広く普及しているので、圧縮したデジタル画像をPDFファイルに格納すると、多くの装置で再生できる。

<16> (10) 上記目的を達成するため、本発明に係る画像圧縮装置は、デジタル画像上に処理対象領域を設定する設定手段と、前記処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化手段と、各前記グループの特徴量に基づいて置換対象グループを選択する選択手段と、前記置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成手段と、前記置換対象グループに属する各画素の色を前記置換対象グループに属さない周囲の画素の色で置換する置換手段と、前記置換手段で前記各画素の色が置換された後の前記デジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮手段と、前記置換対象グループの代表色を表す色データ、前記位置情報、及び前記画像圧縮データを関連付けて出力する出力手段と、を備える。

本発明によると、デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮できる。

<17> (11) 上記目的を達成するため、本発明に係る画像読み取り装置は、原稿を読み取ってデジタル画像を生成する読み取り手段と、請求項10に記載の画像圧縮装置と、を備える。

本発明によると、デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮できる。

<18> (12) 上記目的を達成するため、本発明に係る画像圧縮プログラムは、デジタル画像上に処理対象領域を設定する設定手段と、前記処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化手段と、各前記グループの特徴量に基づいて置換対象グループを選択する選択手段と、前記置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成手段と、前記置換対象グループに属する各画素の色を前記置換対象グループに属さない周囲の画素の色で置換する置換手段と、前記置換手段で前記各画素の色が置換された後の前記デジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮手段と、前記置換対象グループの代表色を表す色データ、前記位置情報、及び前記画像圧縮データを関連付けて出力する出力手段としてコンピュ

一タを機能させる。

本発明によると、デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮できる。

<19> 尚、本発明に備わる複数の手段の各機能は、構成自体で機能が特定されるハードウェア資源、プログラムにより機能が特定されるハードウェア資源、又はそれらの組み合わせにより実現される。また、これら複数の手段の各機能は、各々が物理的に互いに独立したハードウェア資源で実現されるものに限定されない。

<20> 以下、本発明の実施の形態を複数の実施例に基づいて説明する。

#### (第一実施例)

図2は、本発明の第一実施例に係る画像読み取り装置としてのイメージスキャナ1の内部構造を示す模式図である。イメージスキャナ1はパーソナルコンピュータなどによる制御を必要とせず単独で原稿を読み取り可能な所謂スタンドアロン型のイメージスキャナ1であり、原稿を読み取って生成したデジタル画像を圧縮し、圧縮したデジタル画像をPDF (Portable Document Format) 形式でリムーバブルメモリに保存する機能を備えている。

<21> プラテン10はガラス板などの透明板で形成され、上を向く面(原稿載置面)10aに原稿Mが載置される。原稿Mは印刷文書、写真、書籍などの反射原稿、あるいは35mmストリップフィルム、35mmマウントフィルムなどの透過原稿である。

原稿カバー11は、ヒンジ36によりプラテン10の原稿載置面10aを開放する姿勢と原稿載置面10aを覆う姿勢とに揺動可能にハウジング12に連結されている。

<22> 面光源部13は、原稿カバー11に設けられている。面光源部13は、透過原稿用ランプ14、拡散板、反射板等を備える。透過原稿用ランプ14は白色冷陰極蛍光ランプ(CFL)などの管照明装置であり、その長手方向軸線がキャリッジ15のガイドロッド16の中心軸線と平行に延びるように配置されている。面光源部13は、反射原稿の読み取り時に原稿マット35によって覆われる。透過原稿の読み取り時には原稿マット35が外され、面光源部13が露出する。

<23> キャリッジ15は、ハウジング12に收容されている。キャリッジ15は原稿載置面10aに対し平行なガイドロッド16に揺動自在に嵌合している。キャリッジ15は、反射原稿用ランプ17、ミラー18、レンズ19、およびイメージセンサ20を搭載している。キャリッジ15はキャリッジ搬送用モータ33(図3参照)によって駆動されるプーリに掛けられたベルトに牽引され、原稿載置面10aと平行なX方向に往復移動する。反射原稿用ランプ17はCFLなどの管照明装置である。ミラー18およびレンズ19は、反射原稿用ランプ17に照明された反射原稿の反射光および透過原稿用ランプ14に照明された透過原稿の透過光をイメージセンサ20の受光面に入射させる。反射原稿用ランプ17、ミラー18、レンズ19、及びイメージセンサ20がキャリッジ15によってX方向に搬送されると、原稿上の主走査線が移動する。

<24> イメージセンサ20は、フォトダイオードなどで構成される図示しないセルが直線的に多数配列された所謂リニアイメージセンサである。イメージセンサ20は多数のセルが管状の反射原稿用ランプ17の中心軸線と平行に並ぶ姿勢でキャリッジ15に搭載されている。イメージセンサ20の各セルには受光量に応じた電荷が光電変換により蓄積される。各画素に蓄積された電荷は、CCD (Charge Coupled Device) によってイメージセンサ20の出力部に転送される。この電荷転送は、CMOS (Complementary Metal Oxide Semiconductor) によって行われてもよい。イメージセンサ20から出力される画像信号は、AFE部26(図3参照)に出力される。

<25> 図3は、イメージスキャナ1のハードウェアの主要構成を示すブロック図である。

CPU21は、ROM22に記憶されている各種のプログラムを実行してイメージスキャナ1の各部を制御する。また、CPU21はROM22に記憶されている画像圧縮プログラムを実行することにより画像圧縮装置としても機能する。ROM22は各種プログラム、各種データなどを記憶する不揮発性のメモリである。これらの各種プログラムや各種データは、所定のサーバからのネットワークを介したダウンロード、リムーバブルメモリ23などのコンピュータ読み取り可能な記憶媒体からの読み出しなどによってROM22に格納される。RAM24は各種プログラム、各種データなどを一時的に記憶するメモリである。RAMコントローラ25は、CPU21、AFE部26、画像処理部27、外部インタフェースコントローラ28などとRAM24との間のデータ転送を制御する。

<26> 透過原稿用ランプコントローラ29は、インバータ回路、制御回路などから構成され、透過原稿用ランプ14の



点灯、消灯を制御する。

反射原稿用ランプコントローラ30は、インバータ回路、制御回路などから構成され、反射原稿用ランプ17の点灯、消灯を制御する。

イメージセンサコントローラ31はイメージセンサ20を駆動するために必要な駆動パルスイメージセンサ20に出力する駆動回路であり、例えば同期信号発生器、駆動用タイミングジェネレータなどから構成される。

<27> キャリッジコントローラ32は、キャリッジ搬送用モータ駆動回路、制御回路などから構成される。キャリッジコントローラ32は、CPU21から送信されるキャリッジ搬送制御信号を受信し、キャリッジ搬送用モータ33の回転方向、回転速度などを制御する。

アナログフロントエンド(AFE)部26は、電気信号のゲインの調整によるレベル調整処理、量子化処理などを行ってディジタル化されたスキャン画像をRAMコントローラ25を介してRAM24に格納する。

<28> 画像処理部27は、RAM24に記憶されているスキャン画像に対し、ガンマ補正、シェーディング補正、YCbCr色空間で表されるディジタル画像を形成する画像形成処理などの画像処理を施すための信号処理をCPU21と協働して行うDSP(Digital Signal Processor)である。

リムーバブルメモリコントローラ34は、カードコネクタに接続される不揮発性記憶媒体としてのリムーバブルメモリ23にRAM24に格納されているデータを転送する出力機構である。

外部インタフェースコントローラ28は、イメージスキャナ1とPC(Personal Computer)などの外部システムとを通信可能に接続する。

<29> 次に、画像圧縮プログラムについて説明する。

図4は、画像圧縮プログラムの論理的な構成を示すブロック図である。

設定部41は、CPU21を設定手段として機能させるモジュールである。

グループ化部42は、CPU21をグループ化手段として機能させるモジュールである。

選択部43は、CPU21を選択手段として機能させるモジュールである。

位置情報生成部44は、CPU21を位置情報生成手段として機能させるモジュールである。

<30> 置換部45は、CPU21を置換手段として機能させるモジュールである。

画像圧縮部46は、CPU21をディジタル画像圧縮手段として機能させるモジュールである。

出力部47は、CPU21を出力手段として機能させるモジュールである。

<31> 次に、画像圧縮プログラムを実行するCPU21の処理について説明する。

図5及び図6は画像圧縮プログラムを実行するCPU21の処理の流れを示すフローチャートであり、図1は処理の概要を示す模式図である。この処理はイメージスキャナ1が原稿を読み取ってディジタル画像を生成すると開始される。

S105では、CPU21は設定部41を実行し、ディジタル画像を8×8画素のブロックに分割し、ブロック毎にY成分(輝度成分)を離散コサイン変換(DCT変換)することにより、各ブロックを周波数成分に変換する。ブロックはDCT変換により8×8の周波数係数(DCT係数)に変換される。DCT係数において左上は直流成分であり、それ以外は交流成分である。交流成分は左上から右下に向かうほど周波数の高い交流成分(高周波成分)を表している。

<32> 図7は、イメージスキャナが文書を読み取って生成したディジタル画像の一例を示す模式図である。図8は、図7に示すディジタル画像をDCT変換した画像を示す模式図であって、DCT係数の高周波成分が所定値以上であるブロック(以下、「高周波成分を含むブロック」という)のみを黒い四角で示した模式図である。

図5に示すS110では、CPU21は設定部41を実行し、DCT変換したディジタル画像全体を処理対象領域として初期設定する。

<33>

S 1 1 5では、CPU 2 1は設定部 4 1を実行し、設定されている処理対象領域に対して領域分割処理を実行する。領域分割処理では、水平分割処理と垂直分割処理とを繰り返すことにより、ディジタル画像上に複数の矩形の処理対象領域を設定する。

始めに、水平分割処理について説明する。水平分割処理では高周波成分を含むブロックの数が第一の閾値より少ない行を「区切り行」とし、処理対象領域を「区切り行」より上の処理対象領域と下の処理対象領域とに水平に分割する。「区切り行」自体は処理対象領域から除外される。例えば図 9 に示す例の場合、第一の閾値を 3 とすると、処理対象領域 5 1 の第 3 行は高周波成分を含むブロックの数が 2 であるため「区切り行」として特定される。この結果、処理対象領域 5 1 は第 1 行～第 2 行からなる処理対象領域 5 2 と、第 4 行～第 6 行からなる処理対象領域 5 3 とに分割される。ここで第一の閾値は処理対象領域の幅に応じて設定される値である。なお、第一の閾値は例えば行を構成するブロックの総数に占める高周波成分を含むブロックの数の割合であってもよく、割合が第一の閾値未満であれば「区切り行」としてもよい。その場合、割合は処理対象領域の幅によらず一定であってもよい。第一の閾値は適宜選択可能な設計事項であり、処理対象領域が適切に設定されるよう調整すればよい。以下、水平分割処理の詳細について説明する。

<34>

図 1 0 は、水平分割処理の流れを示すフローチャートである。

S 2 0 5では、CPU 2 1は処理対象領域の幅が所定の幅以上であるかを判定する。所定の幅以上であれば S 2 1 0に進む。所定の幅未満のときは S 2 6 0に進む。

S 2 1 0では、CPU 2 1は処理対象領域の幅に基づいて重み付け係数と前述した第一の閾値とを設定する。イメージキャナ 1 では読み取りの際にゴミの影響によって本来は高周波成分を含むべきでないブロックが高周波成分を含んでしまうことがあり、それにより区切り行であるべき行が区切り行として特定されないことが起こり得る。このため CPU 2 1 は後述する S 2 3 5 において、ゴミの影響によって高周波成分を含んでしまったブロック（ゴミブロック）であるかを判定する。重み付け係数はその際に使用されるものである。以下、重み付け係数の一例について説明する。

<35>

図 1 1 は、重み付け係数の一例を示す模式図である。図 1 1 (A) では中央の注目ブロックとその周囲 8 ブロックに 1 が設定されている。CPU 2 1 は、高周波成分を含むブロックのブロック値を 1、高周波成分を含まないブロックのブロック値を 0 とし、ブロック毎にブロック値と重み付け係数との積を求め、それらを合計した値（ブロック判定値）が所定の値（ゴミ判定値）未満であれば、当該注目ブロックはゴミブロックであると判定する。周囲に高周波成分を含むブロックが存在しないところに一つだけ高周波成分を含むブロックが存在する場合、当該注目ブロックはゴミブロックである可能性が高い。図 1 1 (A) に示す重み付け係数によれば、例えば注目ブロックが高周波成分を含むブロックであっても、周囲に高周波成分を含むブロックが存在しなければゴミブロックと見なされる。図 1 1 (B) では注目ブロックとその上下左右の 4 ブロックに 1 が設定されている。また、図 1 1 (C) では注目ブロックにのみ 1 が設定されている。図 1 1 (C) に示す重み付け係数では、注目ブロックがゴミブロックであるか否かの判定に周囲のブロックが加味されない。すなわち図 1 1 (C) に示す重み付け係数の場合はブロック判定値を計算する必要はなく、注目ブロックが高周波成分を含んでいれば高周波成分を含むブロックと判定され、高周波成分を含んでいなければ高周波成分を含まないブロックと判定される。例えば注目領域の幅が広ければ一つや二つのゴミがあっても区切り行の特定に大きな影響はないが、注目領域の幅が狭ければ一つのゴミが区切り行の特定に大きく影響してくる。従って例えば注目領域の幅が広いときは図 1 1 (C) の重み付け係数を用いることによりブロック判定値を計算する処理を省略でき、注目領域の幅が狭いときは図 1 1 (A) の重み付け係数を用いることにより区切り行をより正確に特定できる。その中間のときは図 1 1 (C) の重み付け係数を用いればよい。

<36>

図 1 0 に示す S 2 1 5では、CPU 2 1は処理対象領域の第 1 行を注目行に設定する。ここでいう行とは画素を単位とする行ではなくブロックを単位とする行のことであり、ブロックが直線状に並んでいる行のことである。

S 2 2 0では、CPU 2 1は高周波成分を含むブロックの数をカウントするカウンタ N に 0 をセットする。

S 2 2 5では、CPU 2 1は注目行のブロックを一つ選択し、注目ブロックに設定する。本実施例ではブロックは左端から順に選択していくものとする。

S 2 3 0では、CPU 2 1は注目ブロックが高周波成分を含んでいるかを判定する。具体的には例えば図 1 2 において網掛けした 6 個の交流成分 5 5 の値を所定の重み付け係数で重み付けして合計した値が所定の値（高周波判定値）以上であれば、当該ブロックは高周波成分を含むブロックであると判定する。6 個の交流成分 5 5 は特許請求の範囲に記載の高周波成分に相当する。なお、右下の交流成分を基準にどの範囲までの交流成分を高周波成分と定義するかは適宜選択可能な設計事項であり、例えば処理対象領域が最も適切に設定されると判断される範囲を実験により求めてもよいし、例えば高周波成分は右下の交流成分とその上の交流成分、および左の交流成分の計 3 個であってもよいし、右下の交流成分のみを高周波成分としてもよい。いずれにしても右下の交流成分

は高周波成分である。また、重み付け係数、高周波判定値なども適宜選択可能な設計事項であり、処理対象領域が適切に設定されるよう調整すればよい。高周波成分を含んでいればS235に進み、含んでいなければS245に進む。

<37> S235では、CPU21は注目ブロックがゴミブロックであるか否かを判定する。具体的にはS210で設定した重み付け係数に基づいてブロック判定値を算出し、ブロック判定値がゴミ判定値以上であるか否かを判定する。ゴミ判定値以上であればゴミブロックではないと判定しS240に進む。ゴミ判定値未満であればゴミブロックと判定しS245に進む。

<38> S240では、CPU21はカウンタNに1を加算する。

S245では、CPU21は注目ブロックの右側にブロックが存在するか否かを判定する。存在していればS25に戻り、存在していなければS250に進む。

S250では、CPU21は注目行が区切り行であるか否かを判定する。具体的には、注目行に含まれる高周波成分を含むブロックの数（カウンタN）が第一の閾値未満であるか否かを判定する。第一の閾値未満であれば注目行は区切り行であると判定し、S255に進む。閾値以上であればS265に進む。

<39> S255では、CPU21は注目行の一つ上の行までを一つの処理対象領域として設定する。ただし、上に行そのものがなかったり、一つ上の行が区切り行であったりした場合は処理対象領域は設定されない。

S260では、CPU21はS255で設定した処理対象領域に対して垂直分割処理を実行する。

S265では、CPU21は注目行が最終行であるか否かを判定する。最終行でなければS215に戻る。最終行であれば水平分割処理を終了する。

<40> 次に、垂直分割処理について説明する。

図13は、垂直分割処理の流れを示すフローチャートである。垂直分割処理は、行が列、水平が垂直、幅が高さ、第一の閾値が第二の閾値になる以外は水平分割処理と実質的に同一である。垂直分割処理では、S360において水平分割処理を再帰呼び出しする。垂直分割処理から再帰呼び出しされた水平分割処理は、S260において垂直分割処理を再帰呼び出しする。このようにCPU21は処理対象領域の幅が所定の幅未満、あるいは所定の高さ未満になるまで水平分割処理と垂直分割処理とを互いに再帰的に呼び出すことによりデジタル画像上に複数の矩形の処理対象領域を設定する。

<41> 図14は、上述した領域分割処理の結果を示す模式図である。上述した領域分割処理の結果、図8に示すデジタル画像は図14に示すように複数の処理対象領域56に分割される。図14において平行斜線が付されている複数の矩形領域はそれぞれが一つの処理対象領域に相当する。上述した領域分割処理によると、高周波成分が含まれないブロックからなる領域は平行斜線が付されていない矩形領域となる。文字が含まれない領域は高周波成分を含むブロックが存在しないはずなので、平行斜線が付されていない矩形領域となる。文字が含まれない領域はそのまま高圧縮率で圧縮しても問題ない。にもかかわらず文字が含まれない領域が処理対象領域に設定されると処理に無駄が生じる。上述した領域分割処理によると、文字が含まれない領域は処理対象領域から除外されるので、無駄な処理を低減できる。なお、ここではDCT変換後のデジタル画像の処理対象領域を示しているが、処理対象領域は実際にはDCT変換されていない元のデジタル画像上に設定されるものである。以降の説明において「処理対象領域」とは、DCT変換されていない元のデジタル画像上に設定された処理対象領域のことをいう。

<42> 図5に示すS120では、CPU21は処理対象領域を一つ選択する。

S125では、CPU21はグループ化部42を実行し、選択した処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化処理を実行する。以下、グループ化処理の詳細について説明する。

図15は、グループ化処理の流れを示すフローチャートである。グループ化部42は具体的には例えば公知のk-means法によってグループ化する。k-means法は複数の要素を適切にグループ化するための手法である。

<43> S405では、CPU21は当該処理対象領域の色空間を機種に依存しない色空間に変換する。具体的には例えばLab色空間に変換する。なお、XYZ色空間に変換してもよい。

S410では、CPU21はLab色空間を均等なN個の空間に分割する。ここではN=3の場合を例に説明する。N=3の場合、Lab色空間は図16に示すように27個の均等な空間に分割される。なお、図16では各成



分を-128～127の階調値で表す場合を示している。

<44>

S415では、CPU21は27個の空間の頂点の位置で特定される色を仮の代表色とする。N=3のとき頂点の数は $(N+1)^3$ 個(=64個)である。隣り合う頂点間の距離は全て一定であるため、頂点の位置で特定される色を仮の代表色とすることにより、仮の代表色をL a b色空間全体に偏りなく設定することができる。

S420では、CPU21は処理対象領域の各画素について64個の仮の代表色との色差をそれぞれ求め、各画素をそれぞれ色差が最も小さい代表色と同じグループに配属する。色差は、色差を $\Delta E$ 、ある画素の色を $(L_1, a_1, b_1)$ 、ある代表色を $(L_2, a_2, b_2)$ とすると、以下の数1によって求められる。

$$\Delta E = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2}$$

<46>

この結果、64個の仮のグループが形成される。

S425では、CPU21は各グループについて重心を求め、求めた重心を各グループの新しい仮の代表色とする。ここで重心とは、(グループに属する画素のL成分の平均値、グループに属する画素のa成分の平均値、グループに属する画素のb成分の平均値)で表される色のことをいう。

<47>

S430では、CPU21は各グループについて新旧の代表色の色差を求める。

S435では、CPU21は新旧の代表色の色差が所定の閾値以上であるか否かをグループ毎に判定する。一つ以上のグループで新旧の代表色の色差が所定の閾値以上であれば、S420に戻って処理を繰り返す。上述した処理を繰り返すとやがて重心が収束し、いずれのグループも新旧の代表色の色差が所定の閾値未満となる。色差が所定の閾値未満であれば仮の代表色を正式な代表色としてS440に進む。

<48>

S440では、CPU21は処理対象領域が文字列を表す文章領域であるか否かを判定する。文章領域の場合、図17(A)に示すように背景を表す画素が属するグループと文字を表す画素が属するグループとの2つのグループに画素が集中するはずである。従って、画素がいずれか2つのグループに集中していれば、その処理対象領域は文章領域である可能性が高いといえる。逆に、図17(B)に示すように画素が多くグループに散らばっていれば、その処理対象領域はイラストや写真を表す画像領域である可能性が高いといえる。CPU21は、具体的には例えば全体の80%以上が2つのグループに集中していれば2つのグループに集中していると判定してS445に進み、80%未満であればグループ化処理を終了する。なお、80%を判定基準とするか否かは適宜選択可能な設計事項である。また、ここではいずれか2つのグループに集中していれば文章領域であると判定する場合を例に説明したが、文字が2色以上で印刷されていることも多いので、例えば3～4のグループに集中していれば文章領域であると判定してもよい。

S445では、CPU21は当該処理対象領域に文章領域であることを示すフラグを付してグループ化処理を終了する。

<49>

図5に示すS130では、CPU21は当該処理対象領域が文章領域であるか否かをフラグに基づいて判定する。文章領域の場合はS135に進み、文章領域でない場合はS155に進む。

S135では、CPU21は選択部43を実行し、処理対象領域内の各グループの特徴量に基づいて当該処理対象領域における置換対象グループを選択する選択処理を実行する。ここでは特徴量として各グループに属する画素の数を例に説明する。選択部43は、画素数が二番目に多いグループを置換対象グループとして選択する。画素数が二番目に多いグループを選択する理由は、二番目に多いグループに属する画素は文字そのものを表す画素である傾向が強いからである。例えば図7の破線57で示す処理対象領域において最も多い色は文字の背景を表す灰色である。次に多い色は文字そのものを表す白色である。このように一般の文書では画素の数が一番多いのは背景であり、二番目に多いのは文字である。ただしこれは常に成り立つわけではなく、背景あるいは文字が2色以上で印刷されている場合は文字そのものを表す画素の数は一番目であったり三番目であったりすることもある。このように順位は文書によって異なるので一概にはいえないが、平均すれば二番目である確率は他のいずれの順位である確率よりも高いといえる。なお、前述したように文字が2色以上で印刷されている場合もあるので、2番目に多いグループに加え、例えば3番目に多いグループ、4番目に多いグループも置換対象グループとして選択するようにして

もよい。文字が2色以上で表されている場合、複数のグループを選択することで、文字を表す画素が属するグループが置換対象グループとして選択されずにデジタル画像に文字が残ってしまうことを低減できる。

<51> 第一実施例ではデジタル画像を複数の処理対象領域に分割しているが、複数の処理対象領域に分割すると、特徴量が特定の傾向をより顕著に持つようになるという利点がある。例えば黒い色の文字と赤い色の文字とがあり、黒い色の文字が含まれる領域と赤い文字が含まれる領域とはそれぞれ異なる処理対象領域として設定され、文字を表す画素が属するグループはいずれの処理対象領域でも二番目に多いグループになるとする。ところがそれらの領域が一つの処理対象領域として設定されると、一方は二番目に多いグループになったとしても、他方は三番目に多いグループになってしまうことになり、二番目に多いグループのみを置換対象グループとして選択すると、他方のグループに属する画素がデジタル画像上に残ってしまうことになる。処理対象領域を複数に分割すれば、いずれの処理対象領域においても文字を表す画素が属するグループは二番目に多くなり、文字を表す画素が属するグループは二番目に多いという傾向が顕著に表れ、文字をより確実に除去することができる。

<52> S140では、CPU21は位置情報生成部44を実行し、置換対象グループに属する各画素の位置を示す位置情報を生成する。CPU21は、具体的には例えば処理対象領域を構成する画素を置換対象グループに属する画素と置換対象グループに属さない画素とに2値化したビットマップを生成する。以下、ビットマップについて説明する。

<53> 図18(A)はある処理対象領域の模式図であり、図18(B)はビットマップの模式図である。図18(A)では置換対象グループに属する画素のみを黒枠で示している。処理対象領域が $n \times m$ 画素で構成されているのに対し、ビットマップは $n \times m$ ビットで構成されている。ビットマップの各ビットは処理対象領域の各画素に対応しており、置換対象グループに属する画素に対応するビットは1、置換対象グループに属さない画素に対応するビットは0が設定されている。図18(B)では1が設定されているビットのみを黒枠で示している。画素数が2番目に多いグループに加え、例えば3番目に多いグループ、4番目に多いグループも置換対象グループとして選択した場合は、ビットマップはグループ毎に生成する。なお、位置情報はビットマップでなく置換対象グループに属する各画素の座標を示す情報であってもよい。

<54> 図5に示すS145では、CPU21は生成したビットマップをG4、JBIG、Run Lengthなどの1ビットに適した圧縮方法で圧縮し、ビットマップ圧縮データを生成する。

S150では、CPU21は置換部45を実行し、デジタル画像において置換対象グループに属する画素の色を置換対象グループに属さない周囲の画素の色で置換する置換処理を実行する。以下、置換処理の詳細について説明する。

<55> 図19は置換処理を説明するための模式図である。なお、図19は全ての処理対象領域について置換処理を実行した後のデジタル画像を表している。例えば図7の破線57で示す処理対象領域において「ス」という文字を構成する画素は置換対象グループに属する画素であるとする。この場合、「ス」という文字は周囲を灰色の画素に囲まれているので、置換部45は「ス」という文字を表している画素の色を灰色で置換する。他の文字についても同様である。この結果、図19に示すように処理対象領域から文字が除去される。ただし、上述した例は隣接する画素の色が1色のみの場合の例であり、背景が多色の場合は複数の色に周囲を囲まれることになる。この場合は周囲のいずれか一色のみのみで置換してもよいし、画素毎に周囲の直近の画素を特定して置換してもよい。

<56> 図5に示すS155では、CPU21は次の処理対象領域が存在するか否かを判定する。存在していればS120に戻り、存在していなければS160に進む。

図6に示すS160では、CPU21は画像圧縮部46を実行し、色が置換された後のデジタル画像を高圧縮率で圧縮し、画像圧縮データを生成する。具体的には例えばCPU21は高圧縮率で圧縮するパラメータを用いてデジタル画像をJPEG圧縮する。前述したようにデジタル画像からは文字が除去されているので、高圧縮率で圧縮しても文字が読み難くなるという問題は生じない。

<57> S165では、CPU21は出力部47を実行し、図1に示すように各処理対象領域について置換対象グループの代表色を表す色データ、ビットマップを圧縮して生成されたビットマップ圧縮データ、及び置換対象グループに属する処理対象領域の位置を特定するための領域情報をPDF形式のファイル(PDFファイル)に格納するとともに、デジタル画像を圧縮して生成した画像圧縮データを当該PDFファイルに格納し、リムーバブルメモリコントローラ34を制御してリムーバブルメモリ23に記録する。PDFファイルを再生するプログラムは広く普及しているので、圧縮したデジタル画像をPDFファイルに格納すると、多くの装置で再生できる。ここで領域情報とは、具体的には例えば処理対象領域を示す座標である。置換対象グループとして複数のグループを選択した処理対象領域については、色データとビットマップ圧縮データとを置換対象グループ毎に互いに関連付けて格納する。なお、本実施例ではグループ化処理の際の代表色をPDFファイルに格納する際の代表色としているが、PDF

ファイルに格納する際の代表色は必ずしもグループ化処理の際の代表色でなくてもよい。例えばPDFファイルに格納する際は、グループからランダムに選択した画素の色を当該グループの代表色としてもよい。

再生の際には、領域情報に基づいてビットマップとデジタル画像上の領域とを対応付け、ビットマップで特定される画素についてその画素の色を色データで特定される色に置換することにより、文字は滲むことなく再生される。

<57> 以上説明した本発明の第一実施例に係るイメージスキャナ1によると、各グループの特徴量に基づいて置換対象グループを選択する。処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化した場合、文字を表す画素が属するグループは画素数が二番目に多いという傾向がある。従って、特徴量を用いれば文字を表す画素が属するグループを高い確率で特定することができる。文字を表す画素が属するグループを置換対象グループとし、デジタル画像において置換対象グループに属する画素の色を周囲の画素の色で置換すると、デジタル画像から文字を除去することができる。デジタル画像から文字を除去するとデジタル画像の圧縮率は文字の読み易さに影響を与えなくなるので、デジタル画像を高圧縮率で圧縮することができる。一方、ビットマップは各画素の濃度を持つ必要がないので小さなデータ量でよい。従ってPDFファイルのデータ量を小さくできる。よってイメージスキャナ1によると、デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮できる。

#### <58> (第二実施例)

第二実施例の特徴量は、グループに属する小グループの数である。以下、小グループについて図7の破線57で示す処理対象領域を例に説明する。

小グループとは、他のグループに属する画素に周囲を囲まれて同一のグループに属する他の画素と隔離されている画素群のことをいう。例えば破線57で示す処理対象領域において背景を表す画素は全て同一のグループに配属され、文字を表す画素は全て背景を表す画素とは別の同一のグループに配属される。破線57で示す処理対象領域では背景を表す画素のうち隔離されている画素はないので、全体が一つの小グループに相当する。

<59> 一方、例えば「ジ」の文字は、全部で5つの小グループに分かれている。従って文字を表す画素が属するグループには背景を表す画素が属するグループに比べて多くの小グループが属している。このようにカタカナやひらがなが多い処理対象領域では文字を表す画素が属するグループの小グループの数は文字以外を表す画素が属するグループの小グループの数より多いといえる。従って、小グループの数に着目することにより、文字を表す画素が属するグループを高い確率で特定することができる。ただしこれは常に成り立つわけではなく、例えば漢字が多かったり市松模様の背景であったりすると、文字を表す画素が属するグループの方が小グループの数が少なくなることもあり得る。このように小グループの数は文章によって異なるので一概にはいえないが、特にカタカナやひらがなが多く含まれる文章の場合は一番目である確率は他のいずれの順位である確率よりも高いといえる。

<60> 第二実施例はその他の点において第一実施例と実質的に同一である。

なお、上述した複数の実施例では画像圧縮装置としてイメージスキャナを例に説明したが、パーソナルコンピュータに本発明を適用してもよい。また、デジタルカメラやカメラ付き携帯電話などに本発明を適用してもよい。

<62> 1 イメージスキャナ（画像読み取り装置）、21 CPU（画像圧縮装置、設定手段、グループ化手段、選択手段、位置情報生成手段、置換手段、デジタル画像圧縮手段、出力手段）

#### 【請求項 1】

デジタル画像上に処理対象領域を設定する設定段階と、

前記処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化段階と、

各前記グループの特徴量に基づいて置換対象グループを選択する選択段階と、

前記置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成段階と、

前記置換対象グループに属する各画素の色を前記置換対象グループに属さない周囲の画素の色で置換する置換段階と、

前記置換段階で前記各画素の色が置換された後の前記デジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮段階と、

前記置換対象グループの代表色を表す色データ、前記位置情報、及び前記画像圧縮データを関連付けて出力する出力段階と、

を含むことを特徴とする画像圧縮方法。

【請求項 2】

前記設定段階は、

前記デジタル画像を複数のブロックに分割し、各前記ブロックを周波数成分に変換する変換段階と、

各前記ブロックの高周波成分に基づいて前記処理対象領域を設定する領域設定段階と、

を含むことを特徴とする請求項 1 に記載の画像圧縮方法。

【請求項 3】

前記領域設定段階は、

前記デジタル画像全体を前記処理対象領域として設定する段階と、

前記処理対象領域について前記ブロックからなる行毎に高周波成分が所定値以上である前記ブロックの数を集計し、高周波成分が所定値以上である前記ブロックの数が第一の閾値未満の行で前記処理対象領域を分割する段階と

、  
前記処理対象領域について前記ブロックからなる列毎に高周波成分が所定値以上である前記ブロックの数を集計し、高周波成分が所定値以上である前記ブロックの数が第二の閾値未満の列で前記処理対象領域を分割する段階とを含む。

行で前記処理対象領域を分割する前記段階と列で前記処理対象領域を分割する前記段階とを繰り返すことにより前記デジタル画像上に複数の矩形の前記処理対象領域を設定することを特徴とする請求項 2 に記載の画像圧縮方法。

【請求項 4】

前記グループ化段階において、前記処理対象領域を構成する画素を *k-means* 法によってグループ化することを特徴とする請求項 1、2 又は 3 に記載の画像圧縮方法。

【請求項 5】

前記特徴量は前記グループに属する画素の数であり、

前記選択段階において、画素の数が二番目に多い前記グループを前記置換対象グループとして選択することを特徴とする請求項 1～4 のいずれか一項に記載の画像圧縮方法。

【請求項 6】

他の前記グループに属する画素に周囲を囲まれて同一の前記グループに属する他の画素と隔離されている画素群を小グループというとき、前記特徴量は前記グループに属する前記小グループの数であり、

前記選択段階において、前記小グループの数が最も多い前記グループを前記置換対象グループとして選択することを特徴とする請求項 1～4 のいずれか一項に記載の画像圧縮方法。

【請求項 7】

前記選択段階において、複数の前記グループを前記置換対象グループとして選択することを特徴とする請求項 1～6 のいずれか一項に記載の画像圧縮方法。

【請求項 8】

前記位置情報は、前記処理対象領域を構成する画素を前記置換対象グループに属する画素と前記置換対象グループに属さない画素とに二値化した二値画像、及び前記置換対象グループが属する前記処理対象領域の位置を特定するための領域情報であることを特徴とする請求項 1～7 のいずれか一項に記載の画像圧縮方法。

【請求項 9】

前記出力段階は、前記色データ、前記位置情報、及び前記画像圧縮データを一つの PDF ファイルに格納して出力することを特徴とする請求項 1～8 のいずれか一項に記載の画像圧縮方法。

【請求項 10】

デジタル画像上に処理対象領域を設定する設定手段と、

前記処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化手段と、

各前記グループの特徴量に基づいて置換対象グループを選択する選択手段と、

前記置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成手段と、

前記置換対象グループに属する各画素の色を前記置換対象グループに属さない周囲の画素の色で置換する置換手段と、

前記置換手段で前記各画素の色が置換された後の前記デジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮手段と、

前記置換対象グループの代表色を表す色データ、前記位置情報、及び前記画像圧縮データを関連付けて出力する出力手段と、

を備えることを特徴とする画像圧縮装置。

【請求項 11】

原稿を読み取ってデジタル画像を生成する読み取り手段と、

請求項 10 に記載の画像圧縮装置と、

を備えることを特徴とする画像読み取り装置。

【請求項 12】

デジタル画像上に処理対象領域を設定する設定手段と、

前記処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化手段と、

各前記グループの特徴量に基づいて置換対象グループを選択する選択手段と、

前記置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成手段と、

前記置換対象グループに属する各画素の色を前記置換対象グループに属さない周囲の画素の色で置換する置換手段と、

前記置換手段で前記各画素の色が置換された後の前記デジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮手段と、

前記置換対象グループの代表色を表す色データ、前記位置情報、及び前記画像圧縮データを関連付けて出力する出力手段としてコンピュータを機能させることを特徴とする画像圧縮プログラム。

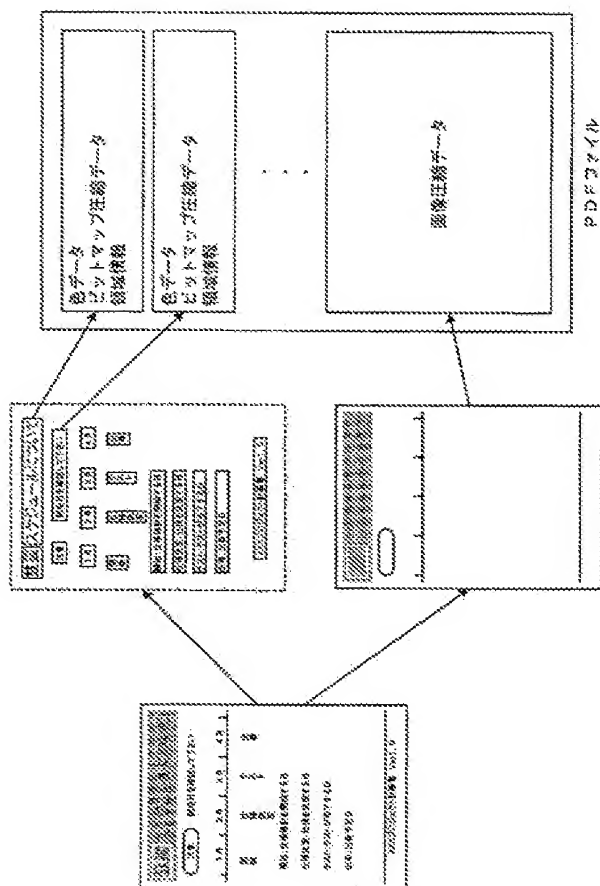


【課題】 デジタル画像を文字の読み易さを損なうことなく高圧縮率で圧縮する画像圧縮方法、画像圧縮装置、画像圧縮プログラム及び画像読み取り装置を提供する。

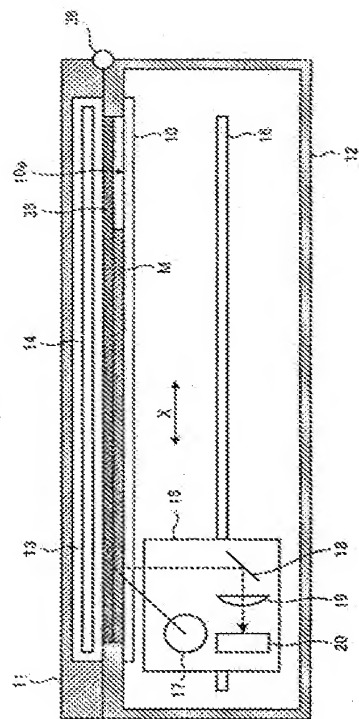
【解決手段】 デジタル画像上に処理対象領域を設定する設定段階と、処理対象領域を構成する画素を互いに色差が小さい画素毎にグループ化するグループ化段階と、各グループの特徴量に基づいて置換対象グループを選択する選択段階と、置換対象グループに属する各画素の位置を示す位置情報を生成する位置情報生成段階と、置換対象グループに属する各画素の色を周囲の画素の色で置換する置換段階と、各画素の色が置換された後のデジタル画像を圧縮して画像圧縮データを生成するデジタル画像圧縮段階と、置換対象グループの代表色を表す色データ、位置情報、及び画像圧縮データを関連付けて出力する出力段階と、を含むことを特徴とする画像圧縮方法。

【選択図】 図1

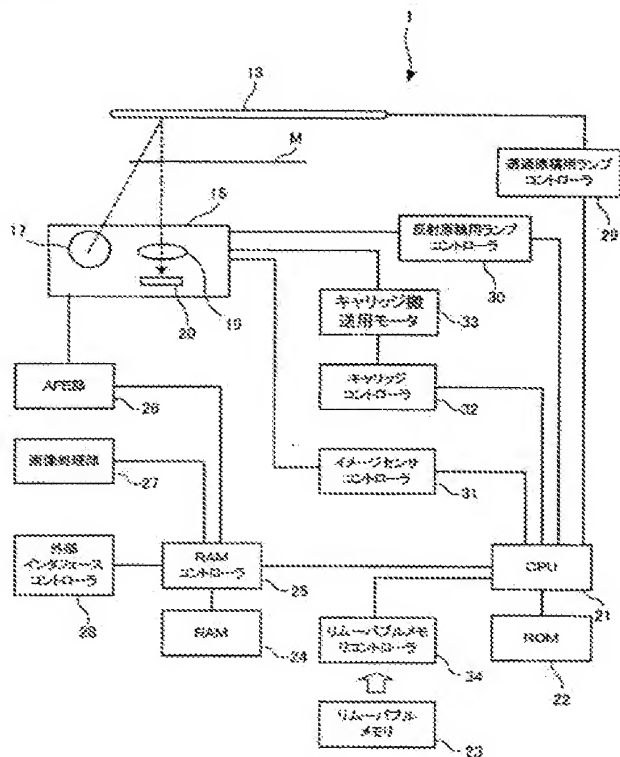
【図1】



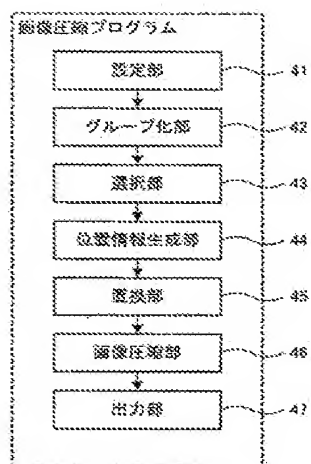
【図 2】



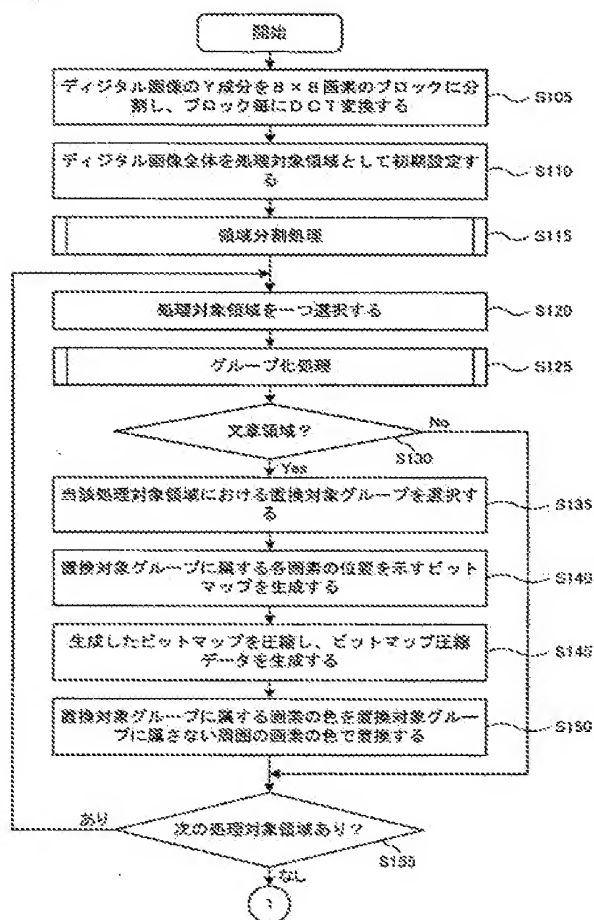
【図 3】



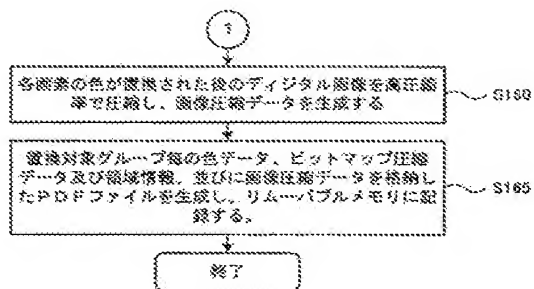
【図 4】



【図 5】



【図 6】



【図 7】

Figure 7 shows a screenshot of a project schedule screen. The screen displays a timeline from January to April, with key milestones marked: 開始 (Start) in January, 仕様決定 (Specification Decision) in February, テスト (Test) in March, and 出荷 (Shipment) in April. Below the timeline, there are definitions for each milestone: 開始: 仕様検討を開始する日 (Start: Day to start specification review), 仕様決定: 仕様を決定する日 (Specification Decision: Day to decide specifications), テスト: テストが完了する日 (Test: Day to complete test), and 出荷: 出荷予定日 (Shipment: Scheduled shipment date). The screen also includes a header with a logo and the text "スケジュール! しまはて", a "注意" (Note) button, and a footer indicating "XXXプロジェクト計画書 Ver1.0".

スケジュール! しまはて

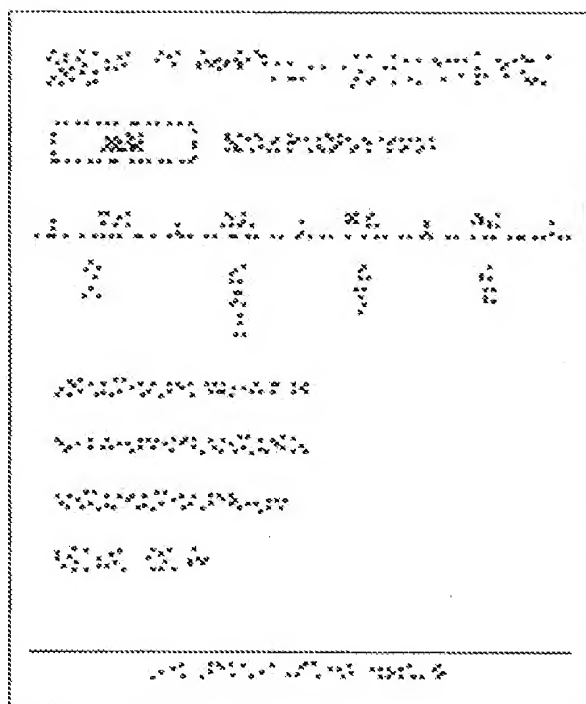
注意 開始日を確認して下さい

1月	2月	3月	4月
開始	仕様決定	テスト	出荷

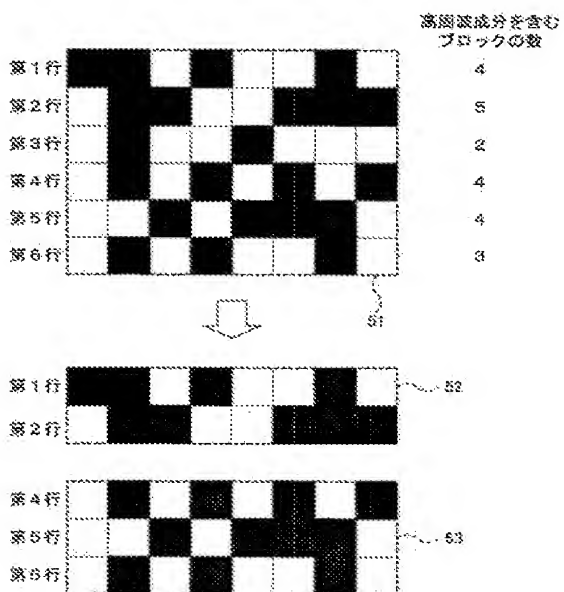
開始: 仕様検討を開始する日  
 仕様決定: 仕様を決定する日  
 テスト: テストが完了する日  
 出荷: 出荷予定日

XXXプロジェクト計画書 Ver1.0

【 8】

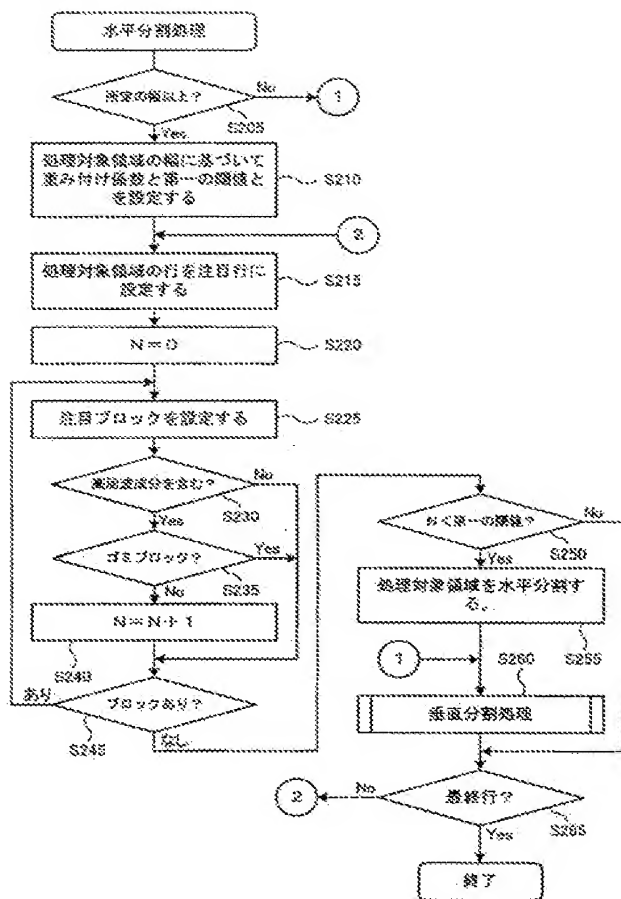


【 9】

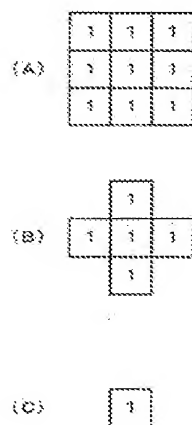




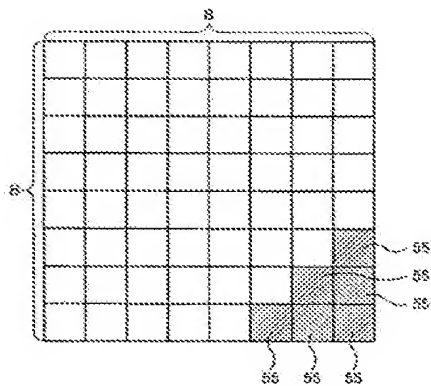
【図 10】



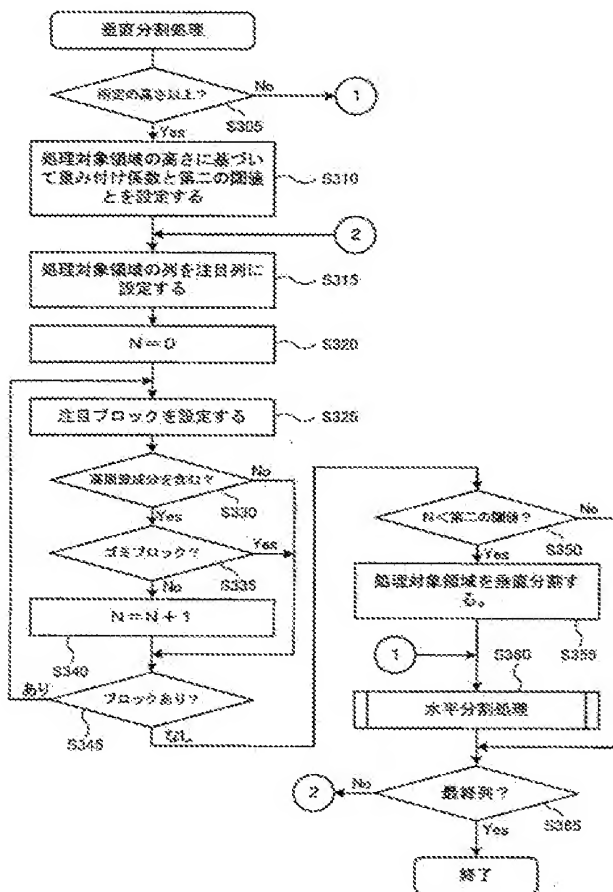
【図 11】



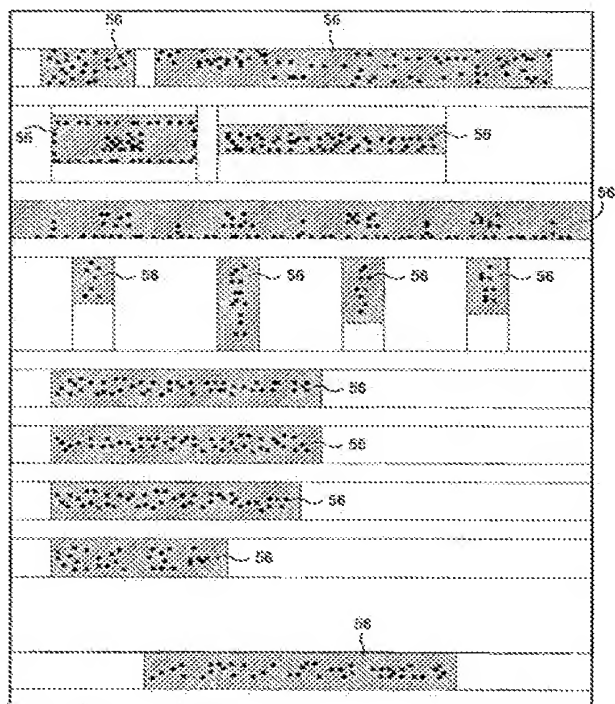
【図 12】



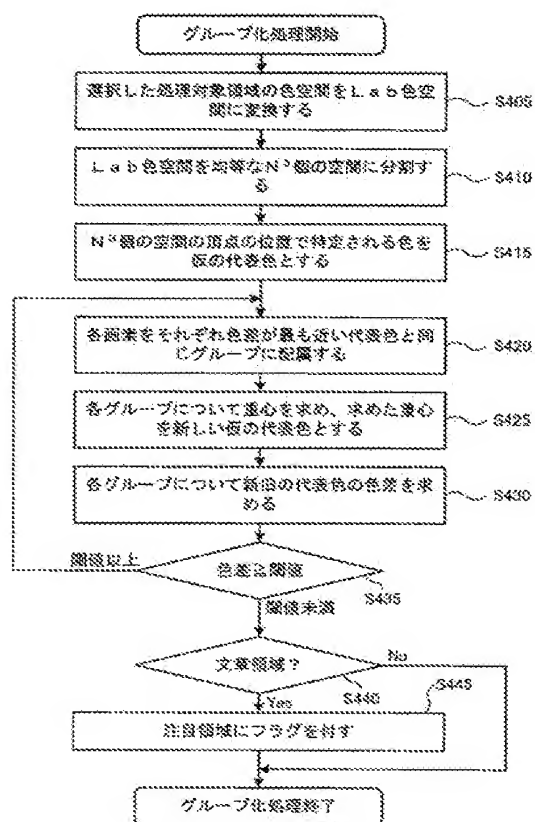
【図 13】



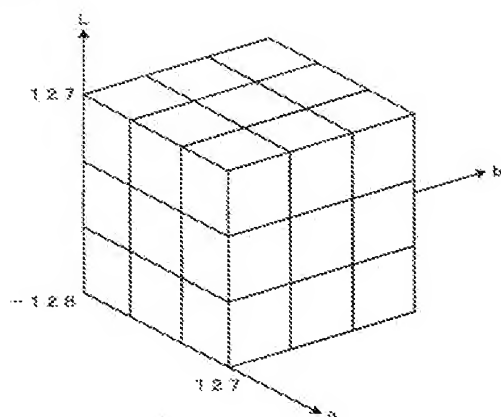
【図 14】



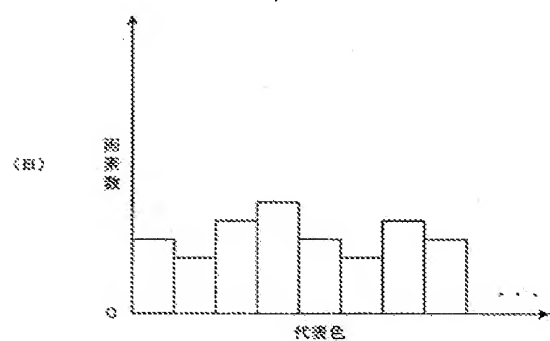
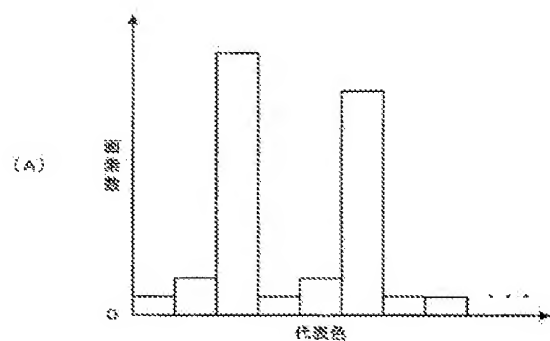
【図 15】



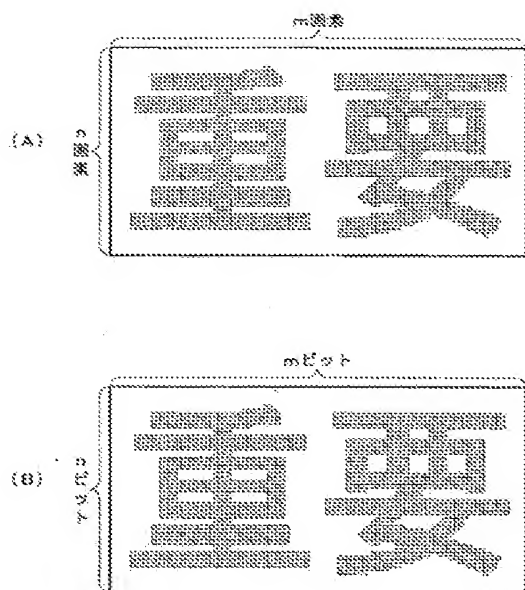
【16】



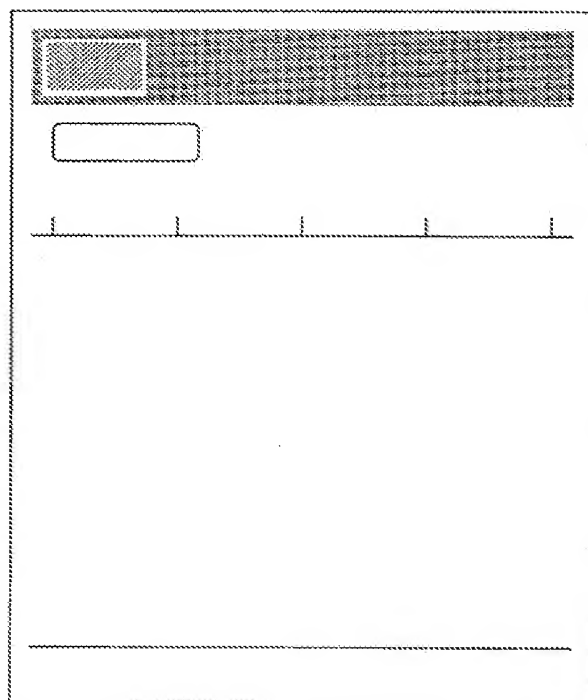
【17】



【図 18】



【図 19】



Fターム(参考) 5C078 AA09 BA42 BA57 CA03 CA22 DA01 DB12



50079 HB08 LA02 LA10 LA27 LA31 LA40 LB11 NA02





## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

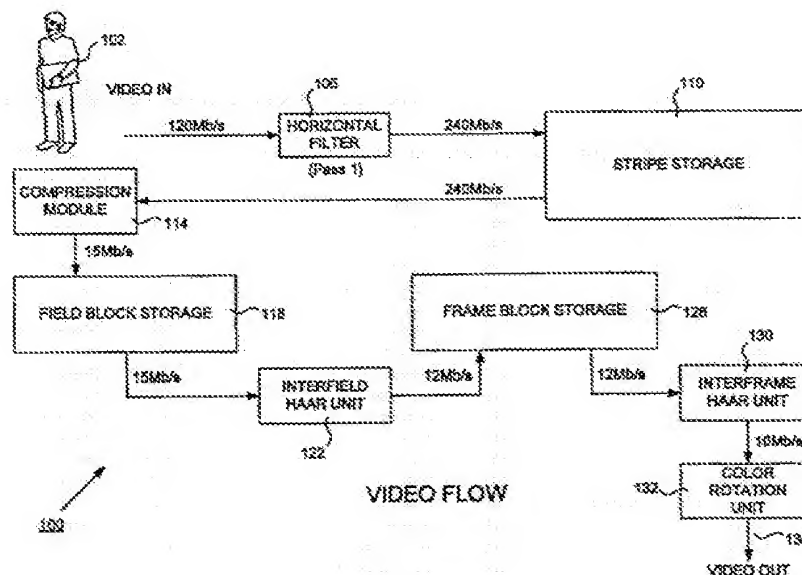
(51) International Patent Classification 6 : <b>H04N</b>		A2	(11) International Publication Number: <b>WO 99/59329</b>
			(43) International Publication Date: 18 November 1999 (18.11.99)
(21) International Application Number: PCT/US99/10688		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 14 May 1999 (14.05.99)			
(30) Priority Data:			
09/079,101	14 May 1998 (14.05.98)	US	
09/079,104	14 May 1998 (14.05.98)	US	
09/079,049	14 May 1998 (14.05.98)	US	
09/079,427	14 May 1998 (14.05.98)	US	
(71) Applicant: INTERVAL RESEARCH CORPORATION [US/US]; Building C, 1801 Page Mill Road, Palo Alto, CA 94304 (US).		<p><b>Published</b></p> <p><i>Without international search report and to be republished upon receipt of that report.</i></p>	
(72) Inventors: LYNCH, William, C.; 3331 Thomas Drive, Palo Alto, CA 94303 (US). KOLAROV, Krasimir, D.; 2050 Avy Avenue, Menlo Park, CA 94025 (US). HOOVER, D., Robert; 16360 Stevens Canyon Road, Cupertino, CA 95014 (US). ARRIGHI, William, J.; 245 San Carlos Avenue, El Cerrito, CA 94530 (US).			
(74) Agent: SCOTT, Jonathan, O.; Beyer & Weaver, LLP, P.O. Box 61059, Palo Alto, CA 94306 (US).			

(54) Title: VIDEO COMPRESSION WITH STORAGE REDUCTION, COLOR ROTATION, COMBINED SIGNAL AND BORDER FILTERING

## (57) Abstract

A technique (200) for compressing video images (102) uses temporary compression of blocks during compression, integrated color rotation of compressed images, direct compression of a composite video signal, and border filters to allow blocks to be compressed independently. Temporary compression reduces storage needed in an integrated circuit. An incoming frame (12) is compressed block-by-block and placed in temporary storage (118). A corresponding block of a later frame is also compressed. Both blocks are decoded back into the transform domain and the two blocks are compared in the transform domain (122). Color rotation (132) on compressed color information is integrated with overall compression and is performed upon the chrominance transform pyramids (592, 596) after transformation of the video signal rather than performing a rotation on the raw signal itself. Color rotation

(132) is performed at any stage and uses serial multiplication (shift and add) for more efficient processing, rather than using parallel multiplication. A composite video signal (82) including both color and black and white information is compressed directly without separating the color information from the black and white. A sequence of passes (214, 224, Figs. 12-16) separates the luminance and chrominance information from the composite video signal and demodulates the color carrier to separate out color information. Blocks of information are treated independently using a modified 2-6 Biorthogonal filter (106a, 409) to reduce complexity, to reduce hardware needed and to reduce blocking artifacts. The technique identifies and compresses composite video, S video, and component video signals, and is applicable to low bit rate video applications.



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TK	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	VU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroun			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

## VIDEO COMPRESSION WITH STORAGE REDUCTION, COLOR ROTATION, COMBINED SIGNAL AND BORDER FILTERING

### FIELD OF THE INVENTION

5           The present invention relates generally to compression and decompression of data. More specifically, the present invention relates to a good quality video codec implementation that achieves a good compression ratio for low bit rate video.

### BACKGROUND OF THE INVENTION

10           A number of important applications in image processing require a very low cost, fast and good quality video codec (coder/decoder) implementation that achieves a good compression ratio. In particular, a low cost and fast implementation is desirable for low bit rate video applications such as video cassette recorders (VCRs), cable television, cameras, set-top boxes and other consumer devices.

15           One way to achieve a faster and lower cost codec implementation is to attempt to reduce the amount of memory needed by a particular compression algorithm. Reduced memory (such as RAM) is especially desirable for compression algorithms implemented in hardware, such as on an integrated circuit (or ASIC). For example, it can be prohibitively expensive to place large amounts of RAM into a small video camera to allow for more efficient compression of images. Typically, smaller amounts of RAM are used in order to implement a particular codec, but this  
20           results in a codec that is less efficient and of less quality.

          Although notable advances have been made in the field, and in particular with JPEG and MPEG coding, there are still drawbacks to these techniques that could benefit from a better codec implementation that achieves a higher compression ratio using less memory. For example, both JPEG and motion JPEG coding perform block-by-block compression of a frame  
25           of an image to produce compressed, independent blocks. For the most part, these blocks are treated independently of one another. In other words, JPEG coding and other similar forms of still image coding end up compressing a frame at a time without reference to previous or



subsequent frames. These techniques do not take full advantage of the similarities between frames or between blocks of a frame, and thus result in a compression ratio that is not optimal.

Other types of coding such as MPEG coding use interframe or interfield differencing in order to compare frames or fields and thus achieve a better compression ratio. However, in order to compare frames, at least one full frame must be stored in temporary storage in order to compare it to either previous or subsequent frames. Thus, to produce the I, B, and P frames necessary in this type of coding, a frame is typically received and stored before processing can begin. The amount of image data for one frame can be prohibitive to store in RAM, and makes such codec implementations in hardware impractical due to the cost and the size of the extra memory needed. In particular, these codec implementations on an integrated circuit or similar device can be simply too expensive due to the amount of memory required.

Previous efforts have attempted to achieve better compression ratios. For example, the idea of performing operations in the DCT transform domain upon a whole frame has been investigated before at UC Berkeley and at the University of Washington for a variety of applications such as pictorial databases (zooming in on an aerial surface map with a lot of detail).

Thus, it would be desirable to have a technique for achieving an improved compression ratio for video images while at the same time reducing the amount of storage needing to be used by the technique. In particular, it would be desirable for such a technique to reduce the amount of memory needed for an implementation on an integrated circuit.

Boundaries between blocks also present difficulties in compression of video images. A brief background on video images and a description of some of these difficulties will now be described. FIG. 1 illustrates a prior art image representation scheme that uses pixels, scan lines, stripes and blocks. Frame 12 represents a still image produced from any of a variety of sources such as a video camera, a television, a computer monitor etc. In an imaging system where progressive scan is used each image 12 is a frame. In systems where interlaced scan is used, each image 12 represents a field of information. Image 12 may also represent other breakdowns of a still image depending upon the type of scanning being used. Information in frame 12 is represented by any number of pixels 14. Each pixel in turn represents digitized information and is often represented by 8 bits, although each pixel may be represented by any number of bits.

Each scan line 16 includes any number of pixels 14, thereby representing a horizontal line of information within frame 12. Typically, groups of 8 horizontal scan lines are organized into a stripe 18. A block of information 20 is one stripe high by a certain number of pixels wide. For example, depending upon the standard being used, a block may be 8x8 pixels, 8x32 pixels, or any other in size. In this fashion, an image is broken down into blocks and these blocks are then transmitted, compressed, processed or otherwise manipulated depending upon the application. In NTSC video (a television standard using interlaced scan), for example, a field of information appears every 60th of a second, a frame (including 2 fields) appears every 30th of a second and the continuous presentation of frames of information produce a picture. On a computer monitor using progressive scan, a frame of information is refreshed on the screen every 30th of a second to produce the display seen by a user.

FIG. 2 illustrates an image 50 that has been compressed block-by-block and then decompressed and presented for viewing. Image 50 contains blocks 52-58 having borders or edges between themselves 62-68. Image 50 shows block boundaries 62-68 having ghosts or shadows (blocking artifacts). For a variety of prior art block-by-block compression techniques, the block boundaries 62-68 become visible because the correlation between blocks is not recognized. Although the block boundaries themselves may not be visible, these blocking artifacts manifest themselves at the block boundaries presenting an unacceptable image.

One technique that is useful for compressing an image block-by-block is to use a 2-6 Biorthogonal filter to transform scan lines of pixels or rows of blocks. A 2-6 Biorthogonal filter is a variation on the Haar transform. In the 2-6 Biorthogonal filter sums and differences of each pair of pixels are produced as in the Haar transform, but the differences are modified (or "lifted") to produce lifted difference values along with the stream of sum values. In the traditional 2-6 Biorthogonal filter, the stream of sum values are represented by the formula:  $s_i = x_{2i} + x_{2i+1}$ , the  $x$  values representing a stream of incoming pixels from a scan line. Similarly, the stream of difference values are represented by the formula:  $d_i = x_{2i} - x_{2i+1}$ . The actual lifted stream of difference values that are output along with the stream of sum values are represented by the formula  $w_i = d_i - s_{i-1}/8 + s_{i+1}/8$ . The 2-6 Biorthogonal filter is useful because as can be seen by the formula for the lifted values "w", each resultant lifted value "w" depends upon a previous and a following sum of pairs of pixels (relative to the difference in question). Unfortunately, this overlap between block boundaries makes the compression of blocks

dependent upon preceding and succeeding blocks and can become enormously complex to implement. For example, in order to process the edges of blocks correctly using the above technique a block cannot be treated independently. When a block is removed from storage for compression, part of the succeeding block must also be brought along and part of the current  
5 block must also be left in storage for the next block to use. This complexity not only increases the size of the memory required to compress an image, but also complicates the compression algorithm.

Prior art techniques have attempted to treat blocks independently but have met with mixed results. For example, for a 2-6 Biorthogonal filter the value of  $w_1$  is calculated using the very  
10 first sum ( $s_0$ ) and the third sum calculated ( $s_2$ ). However, calculation of the very first lifted value ( $w_0$ ) proves more difficult because there is no previous sum with which to calculate the value if the blocks are to be treated independently. The same difficulty occurs at the end of a block when the final lifted value ( $w_{n-1}$ ) is to be calculated, because again, there is no later sum of  
15 pixels to be used in the calculation of this final lifted value if the blocks are to be treated independently. (I.e., a block to be treated independently should not rely upon information from a previous or succeeding block.)

One solution that the prior art uses is to simply substitute zeros for the coefficients (the sum values) in these situations if data values are not known. Unfortunately, this practice introduces discontinuities in the image between blocks and blocking artifacts occur as shown in  
20 FIG. 2. The artifacts occur mainly due to zero values being inserted for some values in the calculation of the initial and final lifted values in the 2-6 Biorthogonal filter. Therefore, it would be desirable for a technique and apparatus that would not only be able to process blocks independently to reduce memory and complexity, but also would do away with ghosts, shadows and other blocking artifacts at block boundaries.

There is a third difficulty associated with processing a video signal which relates to a  
25 color carrier. Color rotation of color information in a video signal typically requires intensive computations. Color rotation is often required to transform a color signal from one coordinate system (or color space) to another. Common coordinate systems are RGB (for television monitors), YIQ (for NTSC television), and YUV (for component video and S video). For  
30 example, for an image that is in the YUV system (as in many drawing programs), a complex

matrix multiplication must be performed to put the image into the RGB system for presentation on a television monitor. Such matrix multiplication requires intensive calculations and larger devices. For example, some color rotations require more computation than all the rest of a compression algorithm, and often a separate semiconductor device is used just to perform the color rotation. Thus, prior art color rotation techniques are relatively slow and costly.

FIGS. 19 and 20 show an example of a prior art color rotation technique. FIG. 19 illustrates frame portions 12a and 12b that represent respectively Y color information and U color information of frame 12. In this example, frame 12 is represented in YUV color coordinates common in component video (Y, or luminance information, not shown). Pixel values  $a(U)$  752 and  $a(V)$  754 represent pixels in corresponding positions of frames 12a and 12b, respectively.

FIG. 20 illustrates a prior art technique 760 for color rotation of information in frame 12 into a different color coordinate system. Each pair of corresponding pixel values 764 (a two entry vector) from frame portions 12a and 12b are multiplied by a rotation matrix R 762 to produce values 766 in the new coordinate system. New values 766 represent the same colors as values 764, but using the different coordinate system. Rotation matrices R have well known values for converting from one coordinate system to another and are 2x2 matrices for converting to YIQ or YUV. Conversion to RGB requires a 3x3 rotation matrix (a three-dimensional rotation). Thus, color rotation requires either two or three multiplications *per element* (per pixel) of a frame. The sheer number of these multiplications make color rotation slow and expensive. Also, the pixel coefficients can be quite large, further intensifying the computations. Therefore, it would be desirable to be able to perform color rotation on a signal without requiring the previous amounts of processing power and device sizes needed.

A fourth difficulty in the prior art exists with respect to compressing composite video and S video signals, i.e., signals that combine colors and/or intensity. In the early days of television it was discovered that the frequency spectrum of a black and white video signal had a large number of unpopulated regions or "holes". Based upon this discovery, it was determined that a color carrier of approximately 3.6 MHz could be added to the black and white (intensity) signal that would "fill in" these unpopulated regions in the frequency spectrum of the black and white signal. Thus, black and white signal information could be added to a color carrier to produce a

composite video signal that, for the most part, kept color and black and white information from interfering with one another. Such a composite video signal 82 and a black and white signal 88 is shown in FIG. 3. Typically, the color carrier signal is modulated by splitting it into two phases 84 and 86 (using quadrature modulation) that are 90° out of phase with each other. Each phase carries one color for the color signal. Each phase is then amplitude modulated, the amplitude of each phase indicating the amplitude of its particular color. Combining signals 84, 86 and 88 produces composite signal 82. Using known techniques, the combination of the two color signals from each phase of the color carrier can be combined with the black and white (intensity) signal to provide the third color. In addition, because the human eye cannot detect high frequency color, the color carrier is often band limited meaning that its frequency does not change greatly.

It is also common to sample a composite video signal at four times the color carrier frequency, often about a 14.3 MHz sampling rate. Signal 82 shows sample points 90-96 illustrating a four times sampling rate for the color carrier signal. Such a sampling rate allows both the carrier and its two phases to be detected and measured; thus, the two phases of the color carrier can be separated out.

Prior art techniques have found it difficult to directly compress such a composite video signal 82. Most prior art techniques separate out the color signals from the black and white signal before compression. Thus, signals 84, 86 and 88 must be separated out from composite signal 82 before compression of the composite signal can begin. This separation of color is expensive and time consuming. Not only are three different algorithms typically needed, but extra hardware may be required. Compression in hardware is often made more complex and costly because of the composite signal. One prior art technique separates out the color signal in analog by using passive components outside of the chip that performs the compression. The three different signals are then fed separately to the compression chip, increasing complexity. Alternatively, separation of the color signal can be done on-chip but this requires extremely large multipliers which greatly increase the size of the chip.

Therefore, it would be desirable for a technique that could handle compression of a composite video signal directly without the need for prior separation of signals or excess hardware. It would be particularly desirable for such a technique to be implemented upon an



integrated circuit without the need for off-chip separation, or for large multipliers on-chip. Such a technique would also be desirable for S video and component video. In general, any combined video signal that includes black and white and color information that needs to be separated during compression could benefit from such a technique.

5           The handling of the different types of video in compression is a fifth area in the prior art that could also benefit from improved techniques. There are three major types of video: composite video; S video; and component video. Composite video is single signal that includes the black/white signal with a color carrier. Modulated onto the color carrier are two chrominance signals. S video is a compromise between composite video and component video.  
10       S video has two signals, a Y signal for black and white information and a single chrominance signal. The single chrominance signal is made up of a color carrier with U and V color signals modulated onto the color carrier. Component video contains three separate signals. A Y signal for black and white information, a U signal for chrominance one information and a V signal for chrominance two information. When compression of a video signal is performed on an  
15       integrated circuit in the prior art, the identification of one of the three types of video signals and preprocessing of that signal is performed off-chip. Prior art techniques have yet to devise an efficient compression algorithm on a single chip that is able to identify and to handle any of the three types of video on the chip itself. It would therefore be desirable for a technique and apparatus by which an integrated circuit could itself handle all three types of video signals and  
20       compress each these signals efficiently.

#### SUMMARY OF THE INVENTION

To achieve the foregoing, and in accordance with the purposes of the present invention, an apparatus and technique for compressing video images are disclosed that address the above difficulties in the prior art.

25           A first embodiment of the present invention uses temporary compression of portions of an image during the overall compression of the complete sequence of images to reduce the amount of temporary storage needed. In particular, this embodiment reduces by a factor of ten the temporary storage needed for interfield and interframe transform-based video compression. In one specific implementation of this embodiment, incoming image data is processed and  
30       compressed block-by-block and placed in temporary storage and then decompressed for

comparison with subsequent blocks before the eventual final compression of the information. Temporary block-by-block compression and the temporal compression of these blocks (between frames, for example) not only allows for a reduction in the temporary storage needed, but also takes advantage of the relationship between associated blocks of an image in order to produce a better picture when the information is finally decompressed. Taking advantage of temporal compression also produces a higher compression ratio. In particular, this technique is especially useful for a codec implemented on an integrated circuit such where less temporary on-chip storage is needed and the chip can be made smaller and faster. Implementation of such a powerful codec on a relatively small and inexpensive integrated circuit provides efficient and high quality video compression in a small device such as a camera or other consumer goods.

In a nutshell, this first embodiment compresses data block-by-block before comparing one block of a first image with its corresponding block in the next succeeding image using a Haar transform. The resulting block can then be encoded and output in a more compressed form. Prior art techniques do not utilize the advantage of temporarily compressing a block and storing it while waiting for its corresponding block to be input. For example, in JPEG and motion JPEG compression video images are generally processed block-by-block and blocks are output in compressed form. There is no notion of temporarily storing compressed blocks in order to compare blocks of a previous image with corresponding blocks of a succeeding image. Other compression algorithms such as those used in MPEG do temporarily store blocks in order to compare a block of a frame to its corresponding block in a later frame. However, storage of these blocks on an integrated circuit (or other device) requires an extraordinary amount of memory which makes the device unnecessarily large and provides a disincentive to perform comparison of corresponding blocks. Advantageously, the present invention stores blocks in a compressed form for comparison with corresponding blocks of a later image. Far less memory is needed on the device to store these compressed blocks. Also, less memory bandwidth is needed for transferring these compressed blocks between memory and a processing unit.

In a specific embodiment, a block is transformed, quantized, and encoded before temporary storage in a much compressed form. Later, when a corresponding block from a later frame arrives, the corresponding block is similarly compressed and stored. Next, both blocks are decoded back into the transform domain. Advantageously, it is not necessary to perform the reverse transform on the stored blocks after decoding them. The two blocks may be compared

in the transform domain. Once the two blocks have been compared, the result is encoded and output as a serial bit stream in a greatly compressed form.

This embodiment greatly reduces the resources required in hardware or software for interframe or interfield video compression. The invention allows for the advantageous comparison of frames or fields but obviates the need to temporarily store a complete frame or field. In particular, the benefits achieved include: less temporary storage required (such as less RAM on an ASIC); lower memory bandwidth requirements between temporary storage (fewer pins on a device and/or faster throughput); reduced computations needed for interframe or interfield comparisons; useful with many compression schemes, such as JPEG, MPEG, H.263 and the like, wavelet compression schemes, etc.; may be used with any transform; and may be used with a variety of standards such as progressive scan and interlaced scan. Also, encoding of blocks can be done using any of a wide variety of techniques.

Another important advantage over prior art compression devices is that intensive operations such as motion compensation in MPEG are not performed. Unlike prior art devices such as the ADV601 available from Analog Devices, Inc. that require multipliers, the present invention uses shift and add for computations. The result is a faster technique and less space required. Also, prior art MPEG compression devices that perform intensive motion compensation are much more complex and expensive (dollar-wise) than their corresponding decompression devices. By contrast, compression and decompression in the present invention have similar complexities; a compression device according to the present invention is relatively less complex and less expensive than an MPEG compression device.

As mentioned above, one important advantage is that earlier frames (or fields or blocks) used as predictors can be kept almost entirely in compressed form throughout the whole process, greatly reducing RAM requirements. This is especially advantageous for implementation on an integrated circuit such as an ASIC where storage area can be one-half to two-thirds of the total area of the chip. For example, for interfield comparisons, only a compressed field buffer of approximately 20 Kbytes per field is needed. In this manner, frame buffers can be greatly reduced or avoided altogether. Images can be reconstructed from the compressed data and the differencing performed on that data. As hardware for decoding is relatively inexpensive, four or five frames worth of data could be decoded at one time. In one alternative embodiment,